# CASCADE PROCESSING FOR SPEEDING UP SLIDING WINDOW SPARSE CLASSIFICATION

*Katariina Mahkonen, Antti Hurmalainen, Tuomas Virtanen, Joni-Kristian Kämäräinen*

Department of Signal Processing, Tampere University of Technology, Finland

## ABSTRACT

Sparse representations have been found to provide high classification accuracy in many fields. Their drawback is the high computational load. In this work, we propose a novel cascaded classifier structure to speed up the decision process while utilizing sparse signal representation. In particular, we apply the cascaded decision process for noise robust automatic speech recognition task. The cascaded decision process is implemented using a feedforward neural network (NN) and time sparse versions of a non-negative matrix factorization (NMF) based sparse classification method of [1]. The recognition accuracy of our cascade is among the three best in the recent CHiME2013 benchmark and obtains six times faster the accuracy of NMF alone as in [1].

***Index Terms***— Automatic speech recognition, non-negative matrix factorization, cascade classification, cascade processing

## I. INTRODUCTION

Classification based on sparse representations (SR) [2], originally invented for image processing [3], has raised to be very popular and provides state-of-the-art results in many disciplines. The model is specifically suitable for modeling data that consists of multiple sources. Recent application fields are for example classification of handwritten characters [4], [5], tracking and classification of vehicles in videos [6], MRI image analysis [7] and EEG signal analysis [8]. Some works [2], [9] no less optimize the sparse object representation specifically for classification.

In the field of audio processing SR have been also widely used, for example in audio classification [10], source separation [11] and content analysis [12]. Also, in the recent CHiME 2013 evaluation [13] the best noise-robust automatic speech recognition (ASR) results [1], [14], [15], [16] were achieved using the sparse non-negative matrix factorization (NMF) method of [1] in combination with two other methods. However, the drawback of SR acquired by iterative non-negative matrix factorization (NMF) algorithms, despite the work on faster algorithms [5], [17], is their high computational demand.

On the other hand, in the field of computer vision, for example, in face recognition [18] and in object detection [19], *cascade processing* has been succesfully used to boost the decision process. Whenever the difficulty of the classification task of the input is not known beforehand, the amount of processing can be regulated with a cascade. The simple decisions can be made with less computing while the most sophisticated methods are used at ambiguous cases.

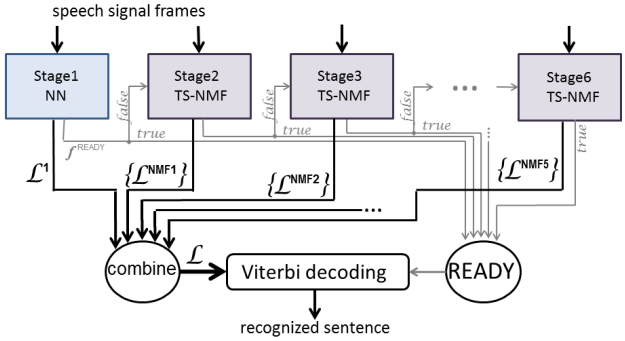


**Fig. 1**. Block diagram of the proposed ASR cascade.

In this work, our aim is to bring spectrogram factorization based noise robust automatic speech recognition closer to real time, while not sacrificing accuracy. Our strategy to reduce computational load is to build a cascade of classifiers (Figure 1), where the amount of computation is determined according to the interpretability of the input. The decision about instantaneous speech content can be made with simple classifiers if the certainty of the estimate is high enough. Estimation certainty assessment in automatic speech recognition has been studied e.g. in [20] and [21], but we propose a simple probability score (section III-C). For our cascade, we develop a time sparse version (TS-NMF) of the NMF method of [1]. We present also an evenly time sparse NMF (ETS-NMF) as a comparison to the cascade structure.

## II. DECISION CASCADE

A decision cascade (DC) for a classification task constitutes of multiple stages where on each the confidence on the input class is evaluated and the decision about completion of the recognition process can be made. This stage-wise processing accounts for the high computational savings that are possible with a DC. A DC is able to preserve the recognition accuracy while at the same time evading redundant computation via early decisions. The effectivity of a DC results from the fact that the easily distinguishable

inputs can be recognized with less processing, i.e. with fast classifiers, while heavier and the most accurate methods need to be executed only for the most ambiguous inputs. The general cascade decision process for classification is presented in Algorithm 1.

There are two functions of special importance within the algorithm, namely $f_s^{\text{READY}}(I) \in \{\textit{false}, \textit{true}\}$ and $f_s^{\text{CLASS}}(I) \in \mathbf{C}$ (class labels). $f_s^{\text{READY}}$ is used to decide whether the decision is ready at stage $s$, and $f_s^{\text{CLASS}}$ gives the class prediction at the stage $s$.

**Algorithm 1:** Decision cascade of $N$ stages.

**Input**: an item $\mathbf{x}$ to be classified
**Output**: class decision C

1 Set READY $=$ *false*
2 Set $\text{S} = 0$
3 **while** READY $\neq$ *true* $\wedge$ $\text{S} \leq N$ **do**
4 $\quad \text{S} = \text{S} + 1$
5 $\quad \text{C} = f_{\text{S}}^{\text{CLASS}}(\mathbf{x})$
6 $\quad$ READY $= f_{\text{S}}^{\text{READY}}(\mathbf{x})$
7 **return** $C$

## III. PROPOSED SPEECH RECOGNITION CASCADE

In ASR a speech signal is converted to sequence of words. Each word is modeled as sequence of states, and likelihoods $\mathcal{L}_t$ of states are estimated in short frames, indexed by $t$. Due to continuous nature of audio signal, the final class decisions are made following a hidden Markov model of the grammar by the Viterbi algorithm, in contrast to independent classification in Algorithm 1 used in [18] and [19].

The stages of the proposed cascade are used to provide increasingly accurate state likelihood estimates, which are accumulated into a state likelihood matrix $\mathcal{L}^s$. Thus the line 5 of Algorithm 1 is replaced with $\mathcal{L}_t^s = f_s^{\mathcal{L}}(\mathbf{x}_t)$, where $\mathcal{L}_t^s \in \mathbb{R}^{N_c \times 1}$ and $N_c$ is the number of states in the grammar.

The proposed ASR cascade aims at speeding up a computationally intensive, but well performing method based on SR and NMF. The cascade works maximally at 6 stages as shown in Figure 1. The first stage uses a NN and subsequent stages use TS-NMF method up to five times to make $f_s^{\text{READY}} = \textit{true}$ if possible. The order of methods within the cascade is defined by the computation time they need to extract the state likelihood information.

Both methods in our cascade extract spectral features of the audio in a 25 ms frame after every 10 ms.

### III-A. Neural Network classifier

The NN classifier at the first stage of the cascade has a topology of two hidden layers, 200 neurons each, and the output layer with $N_c$ neurons. All the neurons use the sigmoid function. The input to the NN is formed of 40 Mel cepstrum coefficients (MFCCs) and delta MFCCs, together 80 features.

Interpretation of NN-output values as probabilities has been investigated in several works, e.g. [22], [23], [24], but we convert NN outputs $\mathbf{y}_t$ to Bayesian posterior probabilities of states as

$$\mathcal{L}_t^{\text{NN}}(c) = P\big(\mathbf{y}_t(c) \,|\, c\big)\; P(c) \;/\; P\big(\mathbf{y}_t(c)\big), \tag{1}$$

with equal priors $P(c)$. For each class $c \in \mathcal{C}$, a histogram based probability density functions (PDF) $P(\mathbf{y}(c))$ and $P(\mathbf{y}(c)|c)$ are collected from the training data.

### III-B. Time Sparse NMF classifier

The later stages of our decision cascade adapt a time sparse versions (TS-NMF) of the original NMF classifier [1]. The NMF classifier processes the input signal in windows of $T = 20$ frames. Spectral magnitudes from $B = 40$ Mel bands from the $T$ frames of a window make an input vector $\mathbf{x}_t$ of length $BT$ for NMF classifier.

A dictionary $\mathbf{D} \in \mathbb{R}_+^{(BT)\times N_d}$ of $N_d = 10000$ such example vectors from training material is used for modeling the input as $\hat{\mathbf{x}}_t = \mathbf{D}\mathbf{w}_t$, where $\mathbf{w}_t$ holds non-negative scores of dictionary elements. The scores $\mathbf{w}_t$ are solved iteratively minimizing Kullback-Leibler divergence between $\hat{\mathbf{x}}_t$ and $\mathbf{x}_t$, which is computationally the heaviest part of the method. Half of the example spectrograms in the dictionary are taken from speech content and the other half from the noise part of training data, notated as $\mathbf{D} = \big[\mathbf{D}^{\text{speech}}, \mathbf{D}^{\text{noise}}\big]$.

State likelihood estimation from scores $\mathbf{w}_t$ is done according to equation (2). Each example vector in $\mathbf{D}^{\text{speech}}$ entails state labels for $T$ consecutive frames. The labels are encoded as binary matrices $\mathbf{L}^d \in \{0,1\}^{N_c \times T}$ to allow mapping the scores $\mathbf{w}_t$ to state likelihoods. An NMF state likelihood window $\mathcal{L}_t^{\text{NMF}}$ spans over time points $t \ldots t+T-1$ and is given by

$$\mathcal{L}_t^{\text{NMF}} = \sum_{d=1}^{N_d/2} \mathbf{L}^d \cdot \mathbf{w}_t(d). \tag{2}$$

In this work we are targeting to reduce computational load, while not giving up the accuracy achievable with a computationally heavy method. The NMF in [1] performs the classification with overlapping windows where an NMF window is factorized for each frame $t = 1, 2, 3, \ldots$. For evenly time sparse NMF (ETS-NMF) a new NMF window is factorized at uniformly spaced frame indices, while in TS-NMF the NMF windows to be factorized can be selected freely. When evaluating ETS-NMF, we found out that with sparsity $p = 3$, i.e. factorizing NMF windows for every third $t$, ETS-NMF produces enough state likelihood information to achieve the accuracy of [1]. Thus in our ASR-cascade, the NMF factorization is allowed only for every third $t$.

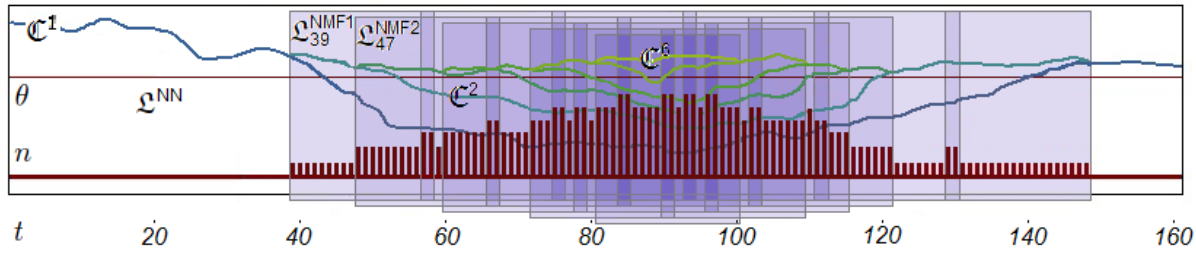


**Fig. 2**. Schema of constructing state likelihoods by NN - TS-NMF cascade processing. The state likelihood matrix $\mathcal{L}^{\text{NN}}$ (white background) is computed at the first stage. The colored curves represent values of $\mathfrak{C}^s$ after each stage $s$. The threshold $\theta$ is shown with the straight line. Where $\mathfrak{C}^s$ does not exceed $\theta$, NMF windows (shaded rectangles) are taken into use by stage $s+1$. Red bars show the value of $n_t$ at each $t$.

### III-C. Cascade decisions

To decide whether the stage $s+1$ should be used to improve state likelihood estimations, the function

$$f^{\text{READY}}(\mathcal{L}^s, t, \theta) = \begin{cases} \mathit{true} & \text{if } \mathfrak{C}^s_t \geq \theta \\ \mathit{false} & \text{else} \end{cases}, \qquad (3)$$

is used. $f^{\text{READY}}$ makes its decisions based on state likelihood matrix $\mathcal{L}^s$ and threshold $\theta$. In (3), $\mathfrak{C}^s_t$ represents the certainty of the information in $\mathcal{L}^s$ at time point $t$ as

$$\mathfrak{C}^s_t = \frac{1}{2l} \sum_{\tau=t-l}^{t+l-1} \max \mathcal{L}^s_\tau$$

The state likelihoods $\mathcal{L}^s_t$ are calculated as a weighted sum of likelihood information acquired from $\mathcal{L}^{\text{NN}}$ given by the NN stage and sets $\left\{\mathcal{L}^{\text{NMF}i}\right\}$ for TS-NMF stages $i = 2 \ldots s$ as

$$\mathcal{L}^s_t = (1 - \frac{n_t}{m}) \cdot \mathcal{L}^{\text{NN}}_t + \frac{n_t}{m} \cdot \left[ \sum_{i=2}^{s} \sum_{\tau=1}^{T} \mathcal{L}^{\text{NMF}(i-1)}_{t-\tau+1}(\tau) \right]_{\mathbf{1}},$$

where $[\cdot]_{\mathbf{1}}$ denotes normalization to the $\ell_1$ length 1. $n_t$ is determined by the number of overlapping NMF state likelihood windows at $t$ and $m = 12$ is used, as it gave the best results. The procedure is elucidated in Figure 2.

The selection of points $\tau$ for NMF windows $\mathcal{L}^{\text{NMF}(s-1)}_\tau$ at each TS-NMF stage $s$ is done as follows. First, each interval of $t$ where $f^{\text{READY}}(\mathcal{L}^s, t, \theta) = \mathit{false}$ is enlarged back- and forward by $T/2$ frames to yield target domain intervals for the new NMF windows. For each interval $\tau_\alpha \ldots \tau_\omega$, $J = \lceil (\tau_\omega - \tau_\alpha + 1)/T \rceil$ new NMF window slots $\tau_j, j = 1 \ldots J$, from $U$ unused slots are selected if possible. The $K = U - J$ slots are left unused as evenly distributed as possible. Finally a new set of NMF factorizations is computed to produce the set of state likelihood windows $\left\{\mathcal{L}^{\text{NMF}(s-1)}_{\tau_j} \text{ for } j = 1 \ldots J\right\}$. New NMF state likelihood windows are generated at subsequent stages until $f^{\text{READY}}(\mathcal{L}^s, t, \theta) = \mathit{true}\ \forall\ t$ or the end of the cascade is encountered. In Figure 2 the set $\left\{\mathcal{L}^{\text{NMF1}}_{\tau_j} \text{ for } j = 1 \ldots 6\right\}$ produced at the second stage of the cascade is illustrated as the uppermost row of shaded NMF windows.

### III-D. Utilizing state unions

In the state space of the used grammar there are many states representing the same phone in different words. For the cascade, it is more advantageous to report the likelihood of a phone instead of a designated state among the phonetically similar states. Thus, considering correlations of the NN output on training data and the states' power in discriminating words, we selected 11 groups to be used as unions. States of the grammar, marked as 'word'$_{\text{state}}$, within unions are $\mathcal{U}_1 = \{\text{`}b\text{`}_2, \text{`}v\text{`}_2\}$, $\mathcal{U}_2 = \{\text{`}b\text{`}_3, \text{`}v\text{`}_3, \text{`}p\text{`}_3, \text{`}g\text{`}_3, \text{`}d\text{`}_3\}$, $\mathcal{U}_3 = \{\text{`}c\text{`}_3, \text{`}t\text{`}_3\}$, $\mathcal{U}_4 = \{\text{`}b\text{`}_4, \text{`}v\text{`}_4, \text{`}p\text{`}_4, \text{`}g\text{`}_4, \text{`}d\text{`}_4, \text{`}e\text{`}_4, \text{`}c\text{`}_4, \text{`}t\text{`}_4\}$, $\mathcal{U}_5 = \{\text{`}a\text{`}_4, \text{`}j\text{`}_4, \text{`}k\text{`}_4\}$, $\mathcal{U}_6 = \{\text{`}i\text{`}_4, \text{`}z\text{`}_2\}$, $\mathcal{U}_7 = \{\text{`}m\text{`}_1, \text{`}n\text{`}_1\}$, $\mathcal{U}_8 = \{\text{`}m\text{`}_4, \text{`}n\text{`}_4\}$, $\mathcal{U}_9 = \{\text{`}f\text{`}_1, \text{`}s\text{`}_1\}$, $\mathcal{U}_{10} = \{\text{`}g\text{`}_1, \text{`}j\text{`}_1\}$ and $\mathcal{U}_{11} = \{\text{`}q\text{`}_4, \text{`}u\text{`}_4\}$.

In $\mathcal{L}^s_t$ the likelihoods of the states within an union are substituted with the highest of them as

$$\mathcal{L}^s_t(c \in \mathcal{U}_i) = max\left\{\mathcal{L}^s_t(c \in \mathcal{U}_i)\right\}$$

for $i = 1 \ldots 11$. The keyword accuracies of both the NMF- and NN-recognizers outside the cascade when using state unions are reported in the experiments (Table I).

## IV. EVALUATION

The evaluation is done using CHiME2013 automatic noisy speech recognition challenge track 1 data [13], which consists of utterances from 34 speakers in highly non-stationary background of domestic noise. Average SNR varies from $-6$ dB to 9 dB. The spoken sentences have strict grammar with 51 words. The state space used to represent the words is defined by the CHiME2013 challenge baseline system and has 4-10 states per word, $N_c = 250$ states in total. The speciality of this data set is the task of recognizing 'coordinates' composed of a letter and a number, e.g. 'D7'. There are 500 and 600 sentences per each SNR level in the training and evaluation set, respectively. The training data is used for training the NN and picking the example vectors for dictionary $\mathbf{D}$ of NMF. The presented recognition accuracies are achieved with the evaluation data set.

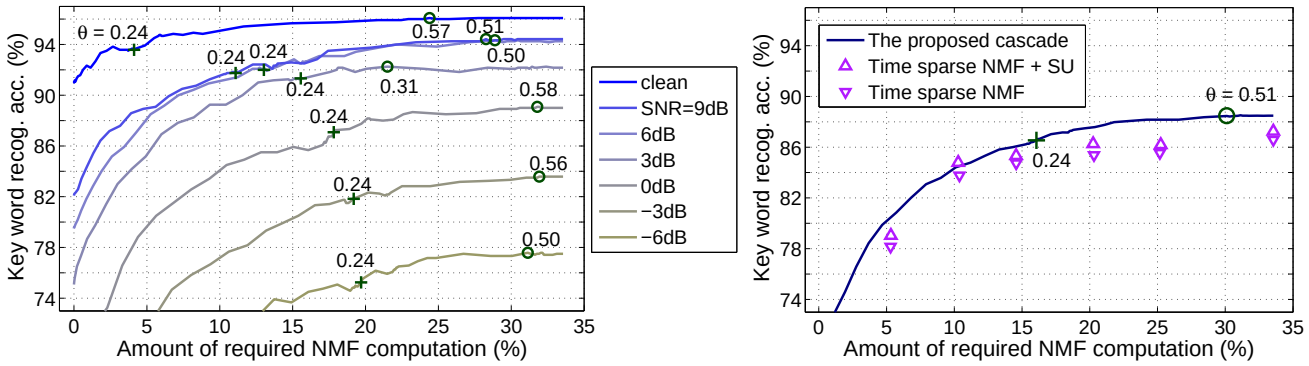


**Fig. 3**. The keyword recognition accuracies of the proposed cascade versus its computational load. The curves build up by changing the threshold $\theta$ of $f^{\text{READY}}$ in eq. (3). The axes on the left show the different SNR levels separately and the average performance is shown on the right. Triangles show average accuracies of ETS-NMF$p$ with sparsities $p = 3, 4, 5, 7, 10, 20$.

### IV-A. Performance with ETS-NMF, NN and state unions

The keyword recognition accuracies on evaluation data with ETS-NMF and the used NN classifier outside the cascade are shown in Table I. The ETS-NMF classifier with time sparsity $p = 3$ utilizing state unions (SU) 'ETS-NMF3+SU' reaches recognition accuracy 87.3 % on average over all the noise conditions. Without SU post processing, 'ETS-NMF3' can be seen to reach the level of the reference 'NMF[1]'. These average accuracies of ETS-NMF3 are shown also as the rightmost triangles in Figure 3. The positive effect of utilizing state unions on ETS-NMF is 0.8 % on average.

The NN classifier of the first stage of the cascade, 'NN+B+SU' in the Table I, reaches accuracy 72.6 % on average. The positive effects of Bayesian post processing (B) of NN outputs and utilizing state unions (SU) are about 1.5 % and 0.9 % respectively.

| SNR | mean | -6dB | -3dB | 0dB | 3dB | 6dB | 9dB |
|---|---|---|---|---|---|---|---|
| ETS-NMF3+SU | 87.3 | 75.4 | 82.4 | 87.8 | 91.3 | 93.0 | 93.5 |
| ETS-NMF3 | 86.6 | 75.1 | 82.0 | 87.4 | 89.9 | 92.3 | 92.8 |
| NMF [1] | 86.5 | 75.6 | 81.4 | 87.5 | 89.9 | 92.4 | 92.3 |
| NN+B+SU | 72.6 | 56.4 | 58.3 | 66.5 | 74.8 | 79.3 | 82.1 |
| NN+B | 71.7 | 55.0 | 57.5 | 65.9 | 73.8 | 78.6 | 81.1 |
| NN | 70.2 | 54.5 | 54.8 | 63.7 | 71.2 | 77.8 | 79.8 |

**Table I**. Keyword recognition accuracies with ETS-NMF3 and NN classifiers with and without using state unions (SU) and Bayesian post processing (B).

### IV-B. Accuracy and computational load of the cascade

The operating point of the proposed cascade is defined by the threshold $\theta$ of $f^{\text{READY}}$ in (3), which rules the usage of stages of the cascade. The threshold $\theta$ is set to achieve a desired accuracy with as small computational load as possible, or to reach as good accuracy as possible with the available computation power. Curves of keyword recognition accuracy, resulting from giving different values for $\theta$, versus the amount of needed NMF computation as percentage of the computational load of [1] are shown in Figure 3. On these curves we pay attention specifically to two operating points. The first one, shown with a cross on each curve, is the operating point with $\theta = 0.24$. It is where the average accuracy reaches 86.5 %, the accuracy of the original NMF framework [1] requiring only 16.0 % of its computation. The second crucial operating point of the cascade, which is shown as a circle on each curve, is where the maximal keyword recognition accuracy is reached with smallest computation load. On average over all noise levels, this operating point occurs with $\theta = 0.51$ reaching accuracy 88.5 % and requiring the computation of 31 % of NMF frames.

### IV-C. Comparison to state-of-the-art

The recognition accuracy of the proposed cascade ranks among the three best in CHiME 2013 challenge Track 1 results in [25]. However, an important aspect of required computational resources was not considered in CHiME 2013 evaluation. Thus in Table II we compare the results with the proposed cascade in comparison to the methods for which we can estimate the computational load: the NMF method of [1] and the winning method [26] of CHiME2013. The computation time of the CHiME2013 winner is obviously higher than NMFs as NMF [1] is one of the three methods in the winning classifier combination.

| | accuracy | computation time |
|---|---|---|
| CHiME2013 winner [26] | 92.8 | $>$ 100 % $^*$) |
| Proposed cascade at $\theta = 0.510$ | 88.5 | 30.9 % |
| ETS-NMF3 | 87.3 | 33.6 % |
| Proposed cascade at $\theta = 0.237$ | 86.5 | 16.0 % |
| NMF [1] | 86.5 | 100 % |

**Table II**. Keyword recognition accuracy of the proposed cascade in comparison to the baseline NMF method and the CHiME2013 challenge winning method ($^*$ utilizes the NMF method as one of its three detectors).

## V. CONCLUSIONS

As automatic noisy speech recognition has proved to be hard problem to solve, the most accurate methods currently are far from real time processing. With clean speech simpler methods might do well, while with noisy environment the more advanced processing is required. A decision cascade is a way to combine these and it is a structure to consider when one wants to meet both the requirements, word accuracy and computational speed, in varying conditions. In this work we have showed that a decision cascade can be successfully applied in ASR task. Our experiments show that the accuracy of well performing NMF method for noisy ASR can be achieved with a fraction of its computation time with a decision cascade utilizing faster classifiers. In CHiME2013 keyword recognition task with our cascade utilizing a neural network and Time Sparse NMF classifiers we achieve the meritorious accuracy of [1] with less than 17 % of its computation time. The full accuracy of the cascade ranks among the three best in CHiME 2013 Track 1 challenge and it is three times faster than the winner.

## VI. REFERENCES


[1] A. Hurmalainen, J. F. Gemmeke, and T. Virtanen, "Modelling non-stationary noise with spectral factorisation in automatic speech recognition," *Computer Speech & Language*, vol. 27, no. 3, 2013.

[2] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Adv Neural Inf Proc Syst*, 2006.

[3] N. Costen, M. Brown, and S. Akamatsu, "Sparse models for gender classification," in *IEEE Int. conf. Automatic Face and Gesture Recognition*, 2004.

[4] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in *Adv Neural Inf Process Syst 28*, 2015.

[5] J. H. Friedman, "Fast sparse regression and classification," *Int J of Forecasting*, vol. 28, no. 3, 2012.

[6] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *TPAMI*, vol. 33, no. 11, 2011.

[7] M. Liu, D. Zhang, D. Shen, A. D. N. Initiative *et al.*, "Ensemble sparse classification of alzheimer's disease," *NeuroImage*, vol. 60, no. 2, 2012.

[8] S. B. Nagaraj, N. Stevenson, W. Marnane, G. Boylan, and G. Lightbody, "A novel dictionary for neonatal eeg seizure detection using atomic decomposition," in *Int Conf on Eng in Medicine and Biology Soc*, 2012.

[9] L. Trottier, B. Chaib-draa, and P. Giguère, "Incrementally built dictionary learning for sparse representation," in *Neural Information Processing*. Springer, 2015.

[10] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, "Shift-invariance sparse coding for audio classification," *Cortex*, vol. 9, 2012.

[11] J. Le Roux, J. R. Hershey, and F. Weninger, "Deep NMF for speech separation," in *In proc ICASSP*, 2015.

[12] C.-T. Lee, yi-hsuan Yang, and H. Chen, "Multipitch estimation of piano music by exemplar-based sparse representation," *IEEE Trans Multimedia*, no. 14, 2012.

[13] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines," in *In proc ICASSP*, 2013.

[14] J. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," in *IEEE TASLP*, vol. 19, 2011.

[15] E. Yılmaz, J. F. Gemmeke, and H. Van hamme, "Noise Robust Exemplar Matching Using Sparse Representations of Speech," *IEEE/ACM TASLP*, vol. 22, 2014.

[16] J. T. Geiger, F. Weninger, J. F. Gemmeke, M. Wöllmer, B. Schuller, and G. Rigoll, "Memory-Enhanced Neural Networks and NMF for Robust ASR," *IEEE/ACM TASLP*, vol. 22, no. 6, 2014.

[17] T. Virtanen, B. Raj, J. F. Gemmeke *et al.*, "Active-set newton algorithm for non-negative sparse coding of audio," in *In proc.ICASSP*, 2014.

[18] P. Viola and M. Jones, "Robust real-time face detection," *Int J Comput Vis*, vol. 57, no. 2, 2001.

[19] T. Wu and S. Zhu, "Learning near-optimal cost-sensitive decision policy for object detection," in *IEEE Int Conf on Comput Vis*, 2013.

[20] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Communication*, vol. 45, no. 4, 2005.

[21] H. Kallasjoki, J. Gemmeke, and K. Palomaki, "Estimating uncertainty to improve exemplar-based feature enhancement for noise robust speech recognition," in *Audio, Speech, and Language Process*, vol. 22, 2014.

[22] J. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," *Neurocomputing NATO ASI Series*, vol. 68, 1990.

[23] M. Richard and R. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, vol. 3, no. 4, 1991.

[24] H. Ney, "On the probabilistic interpretation of neural network classifiers and discriminative training criteria," in *TPAMI*, vol. 17, 1995.

[25] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second chimespeech separation and recognition challenge: An overview of challenge systems and outcomes," in *IEEE workshop on Automatic Speech Recog and Understanding*, 2013.

[26] J. Geiger, F. Weniger, A. Hurmalainen, J. Gemmeke, B. Schuller, G. Rigoll, and T. Virtanen, "The TUM+TUT+KUL approach to the 2nd CHiME challenge: multi-stream ASR exploiting BLSTM networks and sparse NMF," in *The 2nd CHiME workshop on machine listening in multisource environments,*, 2013.